



# DEVICENET (COMMUNICATIONS) REFERENCE







## NOTICE

The products and services described in this manual are useful in a wide variety of applications. Therefore, the user and others responsible for applying the products and services described herein are responsible for determining their acceptability for each application. While efforts have been made to provide accurate information within this manual, Spectrum Controls assumes no responsibility for the accuracy, completeness or usefulness of the information contained herein.

Under no circumstances will Spectrum Controls be responsible or liable for any damages or losses, including indirect or consequential damages or losses, arising out of either the use of any information contained within this manual or the use of any product or service referenced herein.

No patent liability is assumed by Spectrum Controls with respect to the use of any of the information, products, circuits, programming or services referenced herein.

The information contained in this manual is subject to change without notice.

## CAUTION

Spectrum Controls' devices contain electronic components which are susceptible to damage from electrostatic discharge. A static charge can accumulate on the surface of ordinary plastic wrapping or cushioning material. If any Spectrum Controls' device must be returned to Spectrum Controls, the following packaging instruction must be followed:

**PREFERRED:** Use the original packaging material as supplied by Spectrum Controls. Place the device inside the conductive plastic bag.

**ACCEPTABLE:** Wrap the device in some type of antistatic material. Antistatic plastic material can be identified by its pink color, and can be obtained in sheet or bag form.

**UNACCEPTABLE:** Do not use ordinary plastic film, foam, or styrene chips ("popcorn" or "peanuts"). These materials can accumulate charges in excess of 10,000 volts, resulting in possible damage to the Spectrum Controls electronic device.

Antistatic (metallized plastic) bags can be obtained from the following manufacturers:

|               |                |                 |
|---------------|----------------|-----------------|
| 3M Company    | (800-328-1368) | Type 2100 bag   |
| Static, Inc.  | (800-782-8424) | 8000 Series bag |
| Charles Water | (617-964-8370) | CP-303 bag      |



## LIMITED WARRANTY

Spectrum Controls warrants that its products are free from defects in material and workmanship under normal use and service, as described in Spectrum Controls literature covering this product, for a period of 1 year. Spectrum Controls' obligations under this warranty are limited to replacing or repairing, at its option, at its factory or facility, any product which shall, in the applicable period after shipment, be returned to Spectrum Controls' facility, transportation charges prepaid, and which after examination is determined, to the satisfaction of Spectrum Controls, to be thus defective.

This warranty shall not apply to any such equipment which shall have been repaired or altered except by Spectrum Controls or which shall have been subject to misuse, neglect or accident. In no case shall Spectrum Controls' liability exceed the purchase price. The aforementioned provisions do not extend the original warranty period of any product which has either been repaired or replaced by Spectrum Controls.

## Copyright and Trademark information

SOI, SOI-SPS, and SOI-PRO are trademarks of Spectrum Controls, Inc.

IBM is a registered trademark of International Business Machines Corporation.

MS-DOS is a registered trademark of Microsoft Corporation.

DeviceNet is a registered trademark of the Open DeviceNet Vendors Association.

All other trademarks belong to their respective owners.

© 1998 Spectrum Controls, Inc. All rights reserved.  
Specifications subject to change without notice. Printed in U.S.A.



# CONTENTS

|  |    |
|--|----|
| PREFACE  | 1  |
| Who Should Use This Reference .....                  | 1  |
| What This Reference Covers .....                     | 2  |
| Related Documents .....                              | 2  |
| Important Notes .....                                | 2  |
| CHAPTER 1: CABLING AND COMMUNICATION PORTS ...       | 5  |
| Connecting Power .....                               | 6  |
| Uploading and Downloading Application Programs ..... | 6  |
| Connecting to the DeviceNet .....                    | 10 |
| Outputting Printer Forms (SOI-260 only) .....        | 11 |
| Accepting ASCII Input (SOI-260 only) .....           | 11 |
| CHAPTER 2: PROGRAMMING .....                         | 13 |
| Supported File Types .....                           | 13 |
| Supported Data Formats.....                          | 13 |
| Screen Types and Supported Data Formats.....         | 20 |
| Data Display and Entry Screen Parameters.....        | 20 |
| DeviceNet-Specific Configuration Parameters .....    | 26 |
| CHAPTER 3: TROUBLESHOOTING .....                     | 29 |
| General Symptoms and Solutions .....                 | 29 |
| System Status (SOI-260 only) .....                   | 31 |



---

|  |    |
|--|----|
| CHAPTER 4: SPECIAL OPERATIONS .....        | 33 |
| Special Operations on the SOI-120 .....    | 34 |
| Special Operations on the SOI-260 .....    | 39 |
| APPENDIX A: DEVICENET SPECIFICATIONS ..... | 47 |
| Electrical Specifications .....            | 47 |
| Physical Specifications .....              | 47 |
| APPENDIX B: DEVICE PROFILE .....           | 49 |
| DeviceNet Message Types .....              | 50 |
| DeviceNet Object Classes .....             | 51 |
| Identity Object .....                      | 51 |
| Out-Of-Box Definition .....                | 53 |
| Major/Minor Fault Assignments .....        | 55 |
| Message Router Object .....                | 56 |
| DeviceNet Object .....                     | 57 |
| Connection Objects .....                   | 59 |
| Assembly Objects .....                     | 62 |
| Application Object .....                   | 63 |
| INDEX .....                                | 65 |



## PREFACE

Read this preface to familiarize yourself with the rest of this Reference. This preface covers:

- who should use this guide
- what this Reference covers
- related documents
- important notes

Please read all the information in this reference before you install the product. The information contained in this reference applies to hardware and software version 1.0 or later.

### Who Should Use This Reference

Use this Reference if you design, install, program, or maintain a control system that uses a DeviceNet™ network.

This reference assumes a full working knowledge of the relevant programmable controller (PLC) and DeviceNet master (scanner).

You should have a basic understanding of DeviceNet products. If you do not, contact your local distributor for the proper training before using these products.



## What This Reference Covers

This Reference covers the information needed to use the SOI-120 and SOI-260 operator interface products with a DeviceNet network. The Reference contains the information you need to install and program these products. It also provides troubleshooting help and information about the special operations that can be performed with the SOI-120 and SOI-260.

## Related Documents

The following table lists related documents that may help you as you use these products:

| Publication Number | Title                                  |
|--------------------|--|
| 0300051            | SOI-120 Operator Interface User Manual |
| 0300050            | SOI-260 Operator Interface User Manual |
| 0300054            | SOI-SPS Programming Software Manual    |

## Important Notes

### Configuring the DeviceNet master (scanner)

Before you can use an SOI-120 or SOI-260, you must use a DeviceNet configuration tool (such as the Allen-Bradley DeviceNet Manager Software) to configure the DeviceNet master (e.g., scanner).



---

To configure the DeviceNet master:

1. **Add** the SOI-120 or SOI-260 to the list of available devices. Either...
  - **Manually enter** the necessary information (product code, vendor ID, etc.). You can find this information in Appendix B of this reference.

OR

- **Import** an ADS (Electronic Data Sheet) file. We have included an EDS file (on diskette) for the SOI-120 and SOI-260 with the SOI-SPS programming software.
2. **Select** the SOI-120 or SOI-260 (and DeviceNet master, or scanner) from that list.

---

After configuring the DeviceNet master (scanner), you must also configure the SOI-120 or SOI-260 using either the SOI-SPS programming software or the SOI itself (see Chapters 2 and 4, respectively). The settings you enter must match the settings you enter in the DeviceNet configuration tool.

For information on configuring your DeviceNet master (scanner), refer to the documentation that came with your DeviceNet configuration tool.

### Supplying Power to an SOI-120

To create an application program for the SOI-120, you must connect it to a personal computer (see Chapter 1) and supply power to the SOI-120 through its power supply terminal block. See the SOI-120 *Operator Interface Manual* for more information.

**However**, before connecting the SOI-120 to a DeviceNet network, you must *disconnect* power from the SOI-120 power supply terminal block. You should also connect chassis ground to the terminal block.

When connected to a DeviceNet network, the SOI-120 receives power through its DB-9 DeviceNet port and is a non-isolated DeviceNet node.



**CAUTION:** When connecting an SOI-120 to a DeviceNet network, do NOT supply power through the SOI-120's power supply terminal block.



### Supplying Power to an SOI-260

In contrast to the SOI-120, the SOI-260 does not receive power through its DB-9 DeviceNet port.

**However**, before connecting (or disconnecting) the SOI-260 to a DeviceNet network, you must first *disconnect* power from the SOI-260. After connecting the SOI-260 to the DeviceNet network, you may then reconnect power.



**WARNING:** Before connecting or disconnecting an SOI-260 from a DeviceNet network, *disconnect* power from the SOI-260.

---

## CHAPTER 1: CABLING AND COMMUNICATION PORTS

This chapter of the *DeviceNet Communications Reference* covers the cabling needed to use the SOI-120's RS-232/DeviceNet port (see Figure 1.1), as well as the SOI-260's DeviceNet port, optional RS-232/485 communications port, and optional RS-232 printer port (see Figure 1.2).

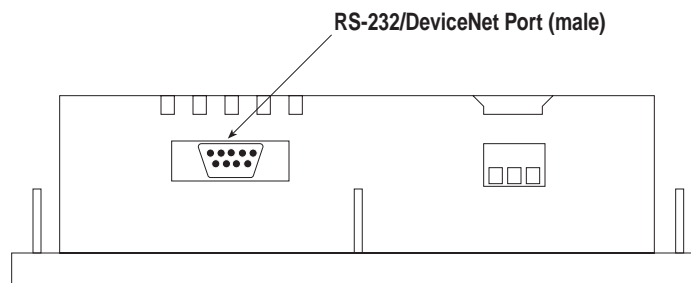


Figure 1.1 SOI-120 Communications Port.

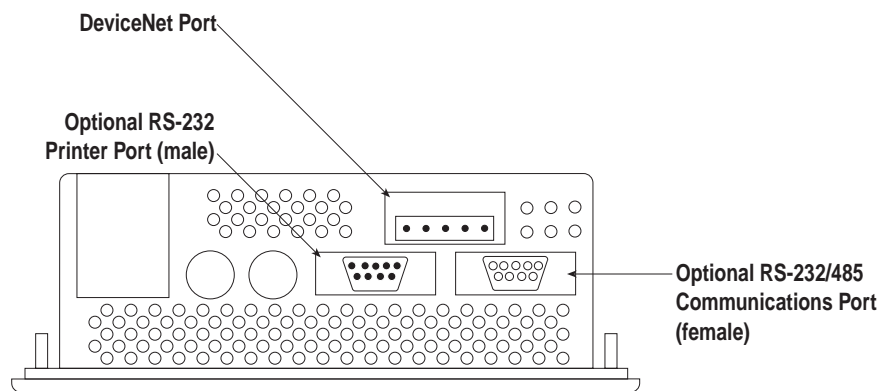


Figure 1.2 SOI-260 Communications Ports.



Use the communication ports to:

- upload or download application programs from a personal computer
- connect your SOI-120 or SOI-260 to the DeviceNet

On the SOI-260 only, you can also use the optional RS-232 printer port to...

- output Printer Forms to a printer or other serial device, such as a large ASCII display unit
- accept data from an ASCII input device (such as a bar code scanner)

## Uploading and Downloading Application Programs

For the SOI-260, you may use either the RS-232 communications port or the RS-232 printer port (whichever is present) to upload or download application programs from a personal computer. For the SOI-120, you use the RS-232/DeviceNet port. The following subsections describe the cabling needed.

### Upload/Download via Communications Port (RS-232 selected)

Figures 1.2 and 1.3 show the cable pin connections (9-pin to 25-pin and 9-pin to 9-pin, respectively) when using the SOI-120's or SOI-260's communications port (RS-232 selected) to upload or download applications from a personal computer. The figures indicate the required connections when building your own cable.

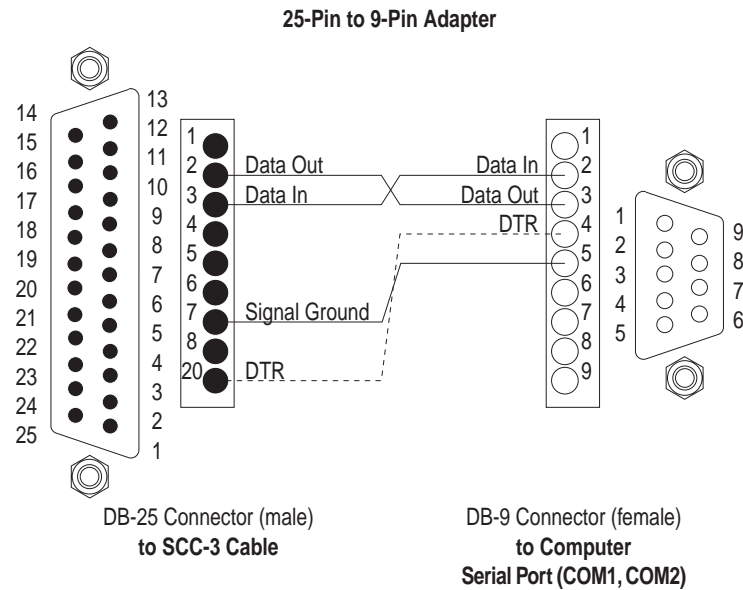


Figure 1.2 Upload/Download cable for 25-pin computer connections.

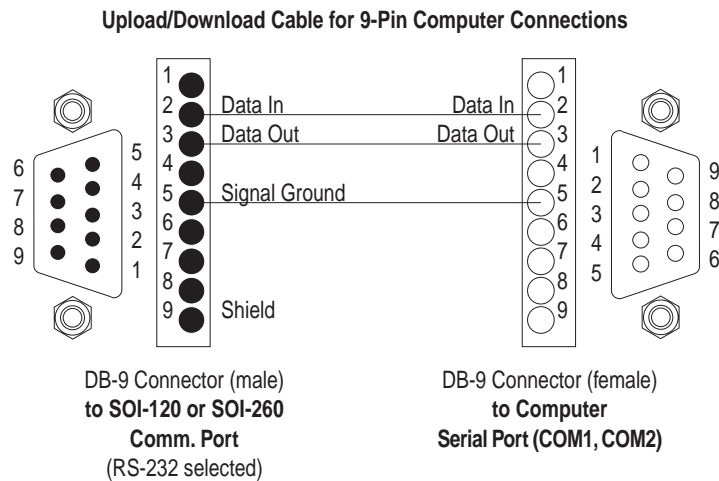


Figure 1.3 Upload/Download cable for 9-pin computer connections.

### Upload/Download via Optional Printer Port (SOI-260 only)

Figures 1.3 and 1.4 show the cable pin connections (9-pin to 25-pin and 9-pin to 9-pin, respectively) when using the SOI-260's optional printer port to upload or download applications from a personal computer. The figures indicate the required connections when building your own cable.

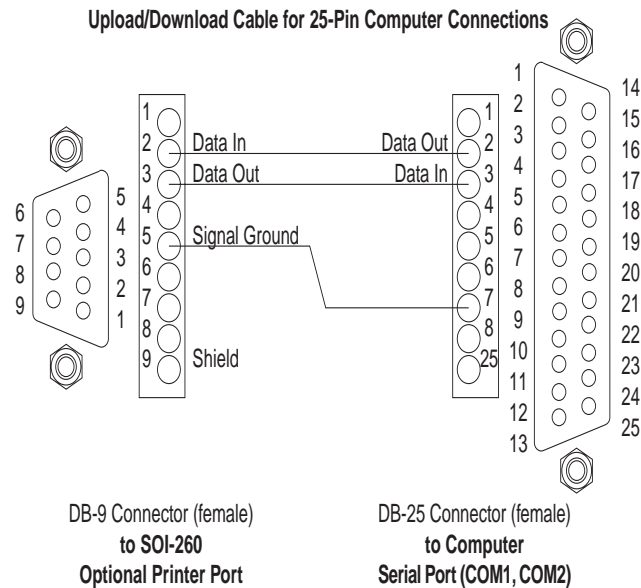


Figure 1.3 Upload/Download cable for 25-pin computer connections.

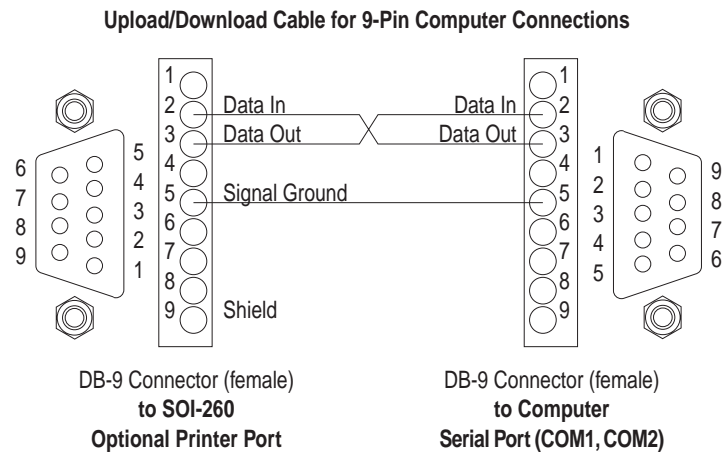


Figure 1.4 Upload/Download cable for 9-pin computer connections.

### Upload/Download with the Spectrum Controls SCC-3 Cable

If you don't want to build your own cable, Spectrum Controls offers the SCC-3 cable for uploading and downloading application programs. The pin connections for the SCC-3 cable are identical to those shown in Figure 1.2. You may use the SCC-3 cable to connect the SOI-120's communications port or SOI-260's communications port or optional printer port to a personal computer. If your computer has a 9-pin communications port, you need a 25-to-9 pin adapter, shown in Figure 1.6. If you want to use the SOI-260's optional printer port for uploading and downloading applications, you also need a 9-pin female adapter, shown in Figure 1.7.

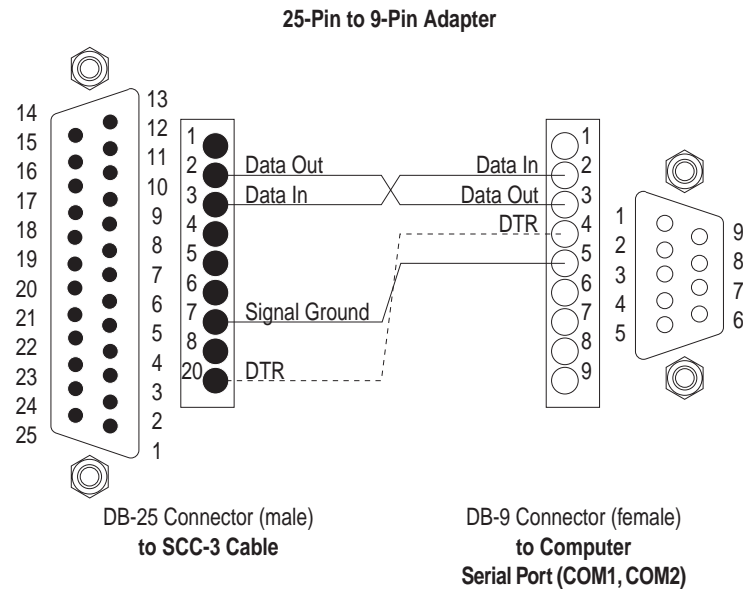


Figure 1.6 25-pin to 9-pin adapter.

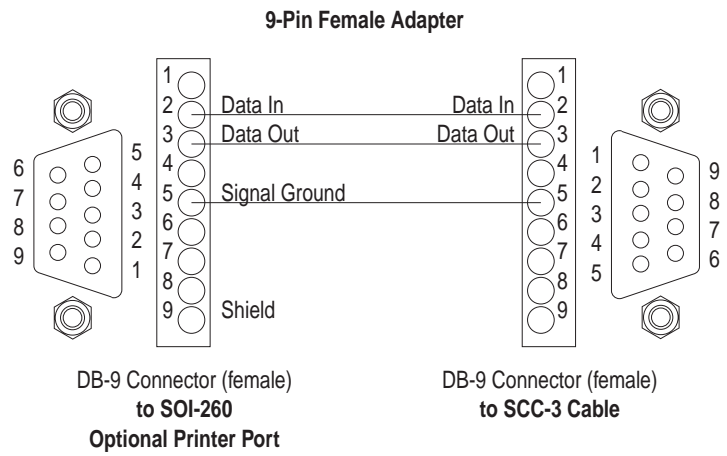


Figure 1.7 9-pin female adapter.

## Connecting to the DeviceNet

The SOI-120 connects to the DeviceNet using a DB-9 connector, shown in Figure 1.8.

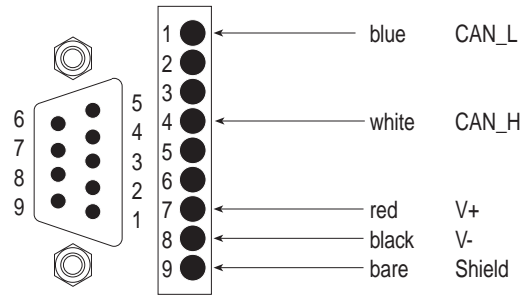


Figure 1.8 DB-9 DeviceNet connector (male) for SOI-120.

The SOI-260 connects to the DeviceNet using a removable terminal block, shown in Figure 1.9.

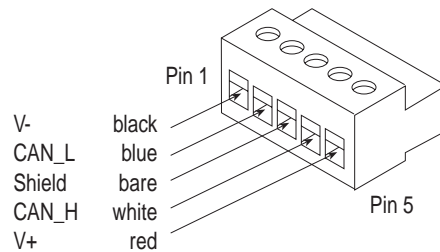


Figure 1.9 Removable terminal block DeviceNet connector for SOI-260.

Devices on end nodes of the DeviceNet require termination. If the SOI-120 or SOI-260 is an end node, you must provide this termination.

To terminate the DeviceNet, install a 121  $\Omega$ , 1%, metal film resistor between the CAN\_H and CAN\_L terminals on the DeviceNet connector, as shown in Figure 1.10

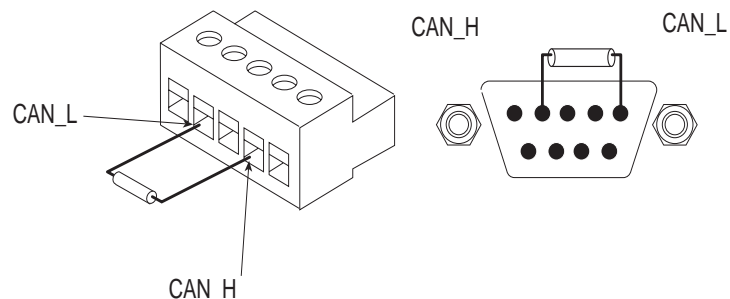


Figure 1.10 DeviceNet termination.

## Outputting Printer Forms (SOI-260 only)

The SOI-260's optional RS-232 printer port may be used to output Printer Forms to a printer or other serial device, such as a large ASCII display unit. Printer forms are created in the SOI-SPS programming software and may include production reports, alarm messages, instructional text, etc.

Figure 1.11 illustrates the cabling needed.

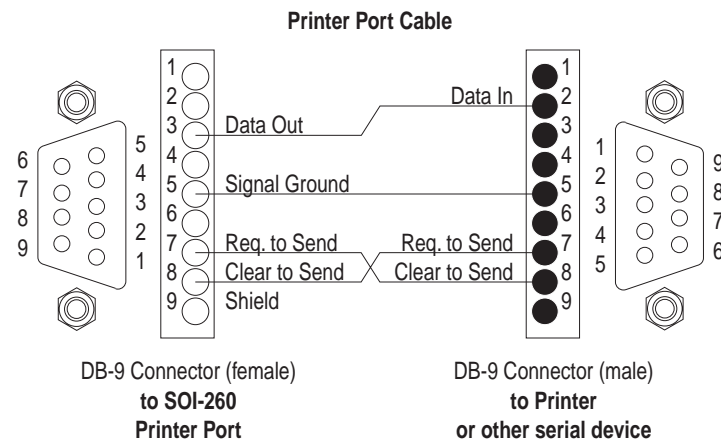


Figure 1.11 Printer port cable.

## Accepting ASCII Input (SOI-260 only)

The SOI-260's optional RS-232 printer port may also be used to accept ASCII data from a variety of devices, including bar code scanners. When using a scanner to read a bar code, the ASCII data is entered directly from the device into a standard data entry screen (created using the SOI-SPS programming software) on the SOI-260.

*Note: If the SOI-260 receives an odd number of characters from the scanner, the SOI-260 adds a Null character to create an even byte count. This facilitates word writes to the controller.*

If keypad entry is enabled, the operator may also enter the decimal equivalent of an ASCII character in the data entry field. The SOI-260 writes data to the controller when any of the following occurs:

- the SOI-260 receives an ASCII carriage return character
- the operator presses the SOI-260's RETURN key
- the SOI-260 receives the maximum number of characters



For example, an operator can manually enter 2 characters and then scan a bar code containing 8 characters, with the last character being an ASCII carriage return. The SOI-260 then writes all 10 characters to the controller.

## CHAPTER 2: PROGRAMMING

This chapter of the *DeviceNet Communications Reference* covers:

- supported programmable controller file types and data formats
- supported SOI-120 and SOI-260 screen types and parameters
- DeviceNet-specific configuration parameters

### Supported File Types

The SOI-120 and SOI-260 support the following two file types:

| File Type | Description | Read/Write Access |
|-----------|-------------|-------------------|
| I         | Input       | Read and Write    |
| O         | Output      | Read only         |

The number of Input and Output words is determined by the Input Buffer Size and Output Buffer Size, which you can modify either from the SOI-120/260 (see “Chapter 4: Special Operations”) or from the SOI-SPS programming software (see the end of this chapter).

### Supported Data Formats

The data format selected for a particular programmable controller memory location must reflect the correct format for the data actually stored in that location. This is the only way you can ensure that correct, consistent information is displayed on the SOI-120 or SOI-260.



For example, selecting a 16-Bit Hex format for one location will display data in one way. Selecting a 16-Bit Signed Integer format for the same location will display the data in another way. It is important to understand each data format and its characteristics.

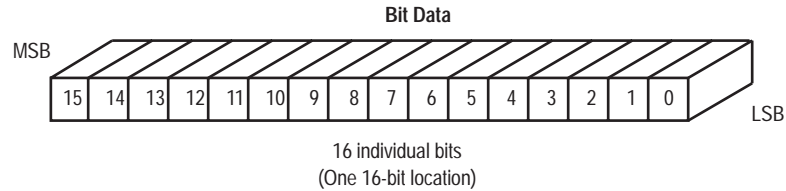
The following table lists all supported data formats and the ranges applicable to each:

| <b>Data Format</b>      | <b>Range</b>                             | <b>Scalable to Engineering Units</b> |
|-------------------------|--|--------------------------------------|
| Bit                     | 0, 1                                     | No                                   |
| 16-bit signed integer   | -32,768 to +32,767                       | Yes                                  |
| 16-bit unsigned integer | 0 to +65,535                             | Yes                                  |
| 16-bit BCD              | 0 to 9,999                               | Yes                                  |
| 16-bit hexadecimal      | 0 to FFFF                                | No                                   |
| 32-bit unsigned integer | 0 to +4,294,967,295                      | No                                   |
| 32-bit BCD              | 0 to +99,999,999                         | No                                   |
| 32-bit hexadecimal      | 0 to FFFF FFFF                           | No                                   |
| 32-bit floating point   | $\pm 1.175495E-28$ to $\pm 3.402823E+28$ | Yes                                  |
| ASCII display           | 0 to 20 characters                       | No                                   |
| ASCII input             | 0 to 32 characters                       | No                                   |

The available data formats are described below.



## Bit

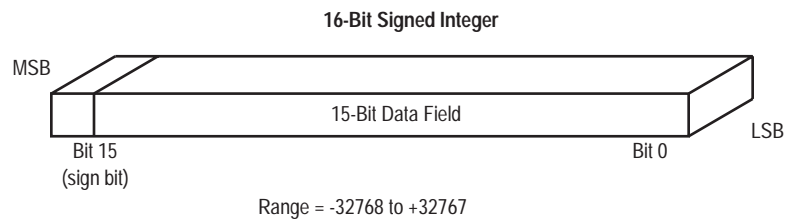


The programmable controller stores a binary (0 or 1) status for a bit location. The SOI will read a programmable controller bit location and determine whether the operational status of the bit is ON (1) or OFF (0). You can customize the text to be displayed on the screen of the SOI-120 or SOI-260 for either state of a specified bit. This text can be up to twenty characters.

For example, the OFF (0) state of a bit might display “Pump is OFF,” and the ON (1) state “Pump is ON.” SOI-SPS allocates enough screen characters for the longest of the two text strings. In this example, 12 characters would be allocated to display “Pump is OFF.”

*Note: The fewer characters used, the less memory is required. In the example above, displaying "OFF" (given the appropriate context) conveys the same information in 3 characters as "Pump is OFF" does with 11 characters.*

## 16-Bit Signed Integer

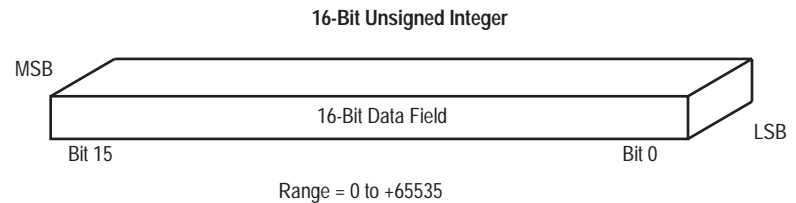


This data type displays a 16-bit register as a signed Integer (two’s complement) value. The 16th bit of the register is the sign bit and is set (1) for a negative and cleared (0) for a positive number. The 16-bit signed integer values have a range of -32768 to +32767.

*Note: This data format may be scaled within the SOI-120 or SOI-260 to different engineering units.*



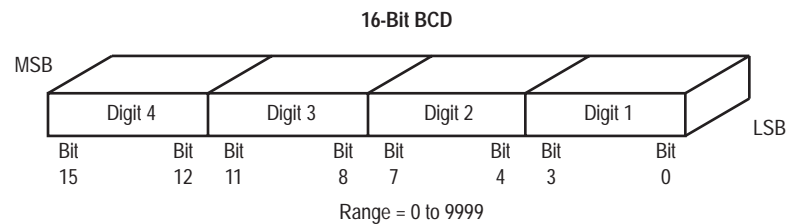
## 16-Bit Unsigned Integer



This data format displays a 16-bit register as an Unsigned Integer value. It represents a positive number by using the 16th bit as a data bit rather than a sign bit. The 16-bit Unsigned Integer values have a range of 0 to +65,535.

*Note: This data format may also be scaled to different engineering units within the SOI-120 or SOI-260.*

## 16-Bit Binary Coded Decimal

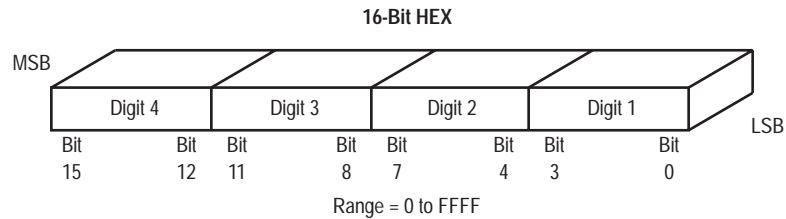


This data type displays a 16-bit register location as a 4-digit Binary Coded Decimal value. The range for the 16 bit BCD selection is 0 to +9999.

*Note: This data format may also be scaled to different engineering units within the SOI-120 or SOI-260.*



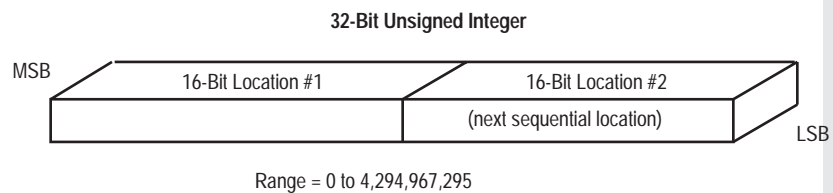
## 16-Bit Hexadecimal



This data type displays a 16-bit register location as a 4-digit hexadecimal value. The range for the 16-bit Hex format is 0 to +FFFF. The Hexadecimal number system is defined as a base of 16 (0-9 and the characters A, B, C, D, E, F).

*Note: This data format may not be scaled to different engineering units within the SOI-120 or SOI-260. It is used for display-only (non-entry) operations.*

## 32-Bit Unsigned Integer

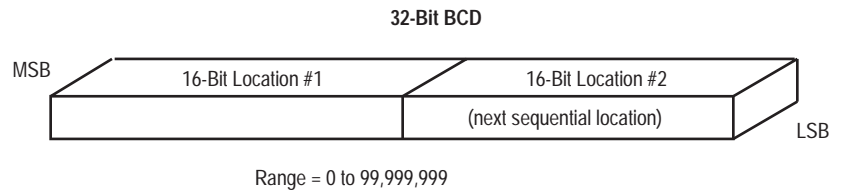


This data format displays data located in two consecutive 16-bit register locations as a 32-bit Unsigned Integer. It uses a memory register plus the next higher register to form the 32-bit location. The High data value is stored in the first register and the Low data value is stored in the next sequential register location. The range for the 32-bit unsigned Integer value is 0 to +4,294,967,295.

*Note: This data format may not be scaled to different engineering units within the SOI-120 or SOI-260.*



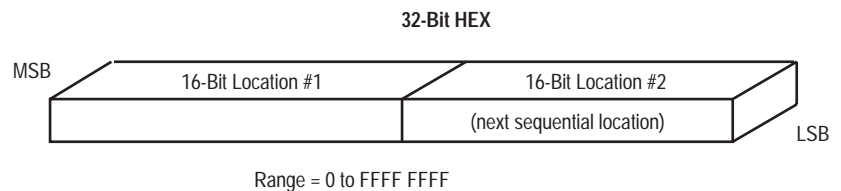
### 32-Bit Binary Coded Decimal



This data type displays two consecutive 16-bit register locations as a 32-bit BCD value. It uses a memory register plus the next higher register to form the 32 bit location. The range for the 32 bit BCD value is 0 to +99,999,999.

*Note: This data format may not be scaled to different engineering units within the SOI-120 or SOI-260.*

### 32-Bit Hexadecimal

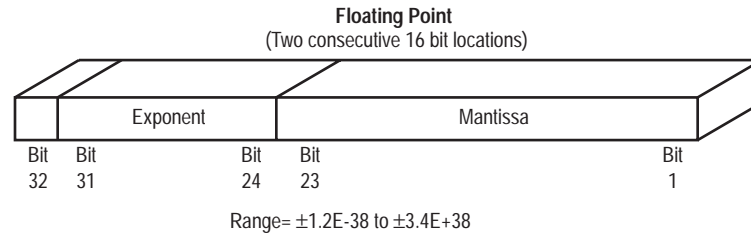


This data type displays two consecutive 16-bit register locations as a 32-bit Hex value. It uses a memory register plus the next higher register to form the 32-bit location. The range for the 32-bit HEX value is 0 to FFFF FFFF.

*Note: This data format may not be scaled to different engineering units within the SOI-120 or SOI-260. This data format is used for display-only (non-entry) operations.*



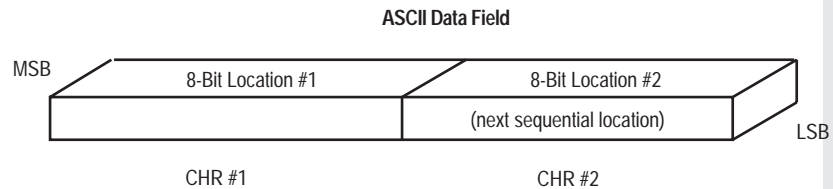
### 32-Bit Floating Point



This data format displays data located in two consecutive 16 bit register locations as a 32 bit floating point value. It uses a memory register plus the next higher register to form the 32 bit location. The range for the floating point value is  $\pm 1.2 \times 10^{-38}$  to  $\pm 3.4 \times 10^{38}$ .

*Note: This data format may be scaled to different engineering units within the SOI-120 or SOI-260.*

### ASCII



Each 16-bit location may contain two ASCII characters (1 byte each). By default, the most significant byte of the base address stores the first character, the least significant byte stores the second character, the first byte of the next sequential location stores the third character, and so on. The data held in this range of address locations is expected to be an ASCII data format.

*Note: The ASCII data format is very useful for programmable controller applications reading ASCII data from bar code readers or data collection terminals.*

*Note: This data format may not be scaled to different engineering units within the SOI-120 or SOI-260.*



## Screen Types and Supported Data Formats

Each SOI Screen type may not support all data formats. The following table lists each screen type and the associated data formats supported:

| Screen Type                       | Data Formats Supported   |
|-----------------------------------|--|
| Display, Alarm, & Printer Form    | Bit<br>16-Bit Signed Integer<br>16-Bit Unsigned Integer<br>16-Bit BCD<br>16-Bit HEX<br>32-Bit Floating Point<br>32-Bit Unsigned Integer<br>32-Bit BCD<br>32-Bit HEX<br>ASCII |
| Entry                             | Bit<br>16-Bit Signed Integer<br>16-Bit Unsigned Integer<br>16-Bit BCD<br>32-Bit Floating Point<br>32-Bit Unsigned Integer<br>32-Bit BCD<br>ASCII Input (SOI-260 only)        |
| Bargraph (SOI-260 only)           | 16-Bit Signed Integer<br>16-Bit BCD  |
| Recipe                            | 16-Bit Signed Integer<br>16-Bit Unsigned Integer<br>16-Bit BCD<br>32-Bit Floating Point<br>32-Bit Unsigned Integer<br>32-Bit BCD   |
| Background Monitor (SOI-260 only) | Bit<br>16-Bit Signed Integer<br>16-Bit BCD   |

## Data Display and Entry Screen Parameters

When programming Data Display and Data Entry fields and positioning them on an SOI-120 or SOI-260 screen, each field must be defined according to its programmable controller memory location (address), data format, and other data parameters specific to the data format selected.



You are prompted for the register definitions at the Control Window. Each register definition is somewhat different, depending on the data format selected and whether the data field is a Data Display or Data Entry field.

The tables in the following two subsections list the applicable data formats and the Control Window prompts associated with each.

### Data Display Screens

The data format selections available for Data Display screens are listed below and their parameters described.

| <b>Display Screen<br/>Parameter for:<br/>Bit Data</b> | <b>Description</b>   |
|---|--|
| Register Number                                       | The programmable controller data location operand and address.   |
| Bit Number  | The Bit number if the register number designates a multiple bit location (a 16 bit data address, for example). |
| Text when Bit is OFF (0)                              | The 20 character text description to be displayed when the bit is in an OFF (0) state                          |
| Text when Bit is ON (1)                               | The 20 character text description to be displayed when the bit is in an ON (1) state.                          |

**Display Screen****Parameter for:****16-Bit Unsigned Integer,****16-Bit Signed Integer &****16-Bit BCD Data****Description**

|                               |  |
|-------------------------------|--|
| Register Number               | The programmable controller data location operand and address.   |
| Digits Right of Decimal       | The number of digits to be placed to the right of the decimal.   |
| Digits Left of Decimal        | The number of digits to be placed to the left of the decimal.  |
| Leave Place for Sign (Y or N) | Leave a one character place for the polarity sign (+ or -) when the data is displayed.   |
| Show Leading Zeros (Y or N)   | Display any zeros to the left of the data.   |
| Minimum Register Value        | The minimum data value of the programmable controller location.  |
| Maximum Register Value        | The maximum data value of the programmable controller location.  |
| Minimum Displayed Value       | The minimum data value to be displayed. This value is displayed when the data in the programmable controller location is equal to the minimum register value. See Maximum Displayed Value, below, for a description of scaling.  |
| Maximum Displayed Value       | The maximum data value to be displayed. This value is displayed when the data in the programmable controller location is equal to the maximum register value. The range defined by the Minimum Displayed Value and the Maximum Displayed Value is proportionally scaled to the range of the minimum and maximum register values. If both ranges are equal then the scaling ratio is 1:1. |
| Minimum Bar Value             | (Bar Graph Only) The minimum value of data to be displayed in the Bar Graph. This value must be greater or equal to the Minimum Displayed Value.   |
| Maximum Bar Value             | (Bar Graph Only) The maximum value of data to be displayed in the Bar Graph. This value must be less than or equal to the Maximum Displayed Value. The Minimum and Maximum bar graph values may be used to display a particular range or window of an overall range (Minimum and Maximum Displayed Values).  |



| Display Screen<br>Parameter for:<br>32-Bit Unsigned Integer &<br>32-Bit BCD Data |  | Description  |
|--|--|--|
| Register Number  |  | The programmable controller data location operand and address. |
| Digits Right of Decimal  |  | The number of digits to be placed to the right of the decimal. |
| Digits Left of Decimal   |  | The number of digits to be placed to the left of the decimal.  |
| Show Leading Zeros (Y or N)  |  | Display any zeros to the left of the data.                     |

*Note: For the 32-Bit BCD selections, scaling of data is not supported. The selected register number and the next higher sequential register number identify the locations defining the 32 bit data value.*

| Display Screen<br>Parameter for:<br>16 -Bit HEX,<br>32-Bit HEX Data |  | Description  |
|---|--|--|
| Register Number   |  | The programmable controller data location operand and address. |

| Display Screen<br>Parameter for:<br>ASCII Data: |  | Description  |
|---|--|--|
| Register Number                                 |  | The programmable controller data location operand and address.   |
| Character Count                                 |  | The number of characters (2 characters for each 16 bit data location) to be displayed, up to a maximum of 20 characters. The initial byte of the location identified by the register number is displayed first, then the second byte, the first byte of the next higher sequential location, and so on. To display 20 characters, a sequential block of ten 16 bit locations is read by the SOL. |



## Data Entry Screens

The data format selections available for Data Entry screens are listed below and their parameters described.

**Entry Screen**  
**Parameter for:**  
**Bit Data**

| Description  | Description   |
|--|---|
| Register Number  | The programmable controller data location operand and address.  |
| Bit Number   | The Bit number if the register number designates a multiple bit location (a 16 bit data address, for example). This selection is irrelevant if the register number refers to a Bit type address.  |
| Input Data by Pressing<br>'1'/'0' or 'Y'/'N' ?<br>(Enter 1 or Y) | This parameter determines whether the operator will enter 1 or Y to set the defined bit location. If 1 is entered, 0 will clear the bit location. If Y is entered, N will clear the bit location.   |
| Default Value ?<br>Z = No Default<br>(Enter 1,0,Y,N,Z)           | This parameter defines the default value that is displayed at the data entry position of the SOI display. If a default value of Y is entered, a Y is displayed, and the operator is only required to press ENTER to set the bit location.<br>An entry of Z defines no default value. If there is no default value programmed, and the operator presses the ENTER, no data is sent to the programmable controller. |

**Entry Screen**  
**Parameter for:**  
**16-Bit Unsigned Integer,**  
**16-Bit Signed Integer &**  
**16-Bit BCD Data**

| Description                   | Description  |
|-------------------------------|--|
| Register Number               | The programmable controller data location operand and address.                         |
| Digits Right of Decimal       | The number of digits to be placed to the right of the decimal.                         |
| Digits Left of Decimal        | The number of digits to be placed to the left of the decimal.                          |
| Leave Place for Sign (Y or N) | Leave a one character place for the polarity sign (+ or -) when the data is displayed. |
| Minimum Register Value        | The minimum data value of the programmable controller location.                        |



|                        |  |
|------------------------|--|
| Maximum Register Value | The maximum data value of the programmable controller location.  |
| Minimum Entry Value    | The minimum data value to be entered. When this value is entered the minimum register value is entered to the defined programmable controller location. See Maximum Entry Value, below, for a description of scaling.  |
| Maximum Entry Value    | The maximum data value to be entered. When this value is entered the maximum register value is entered to the defined programmable controller location. The range defined by the minimum entry value and the maximum entry value is proportionally scaled to the range of the minimum and maximum register values. If both ranges are equal then the scaling ratio is 1:1. |
| Low User Input Limit   | The minimum entry value that an operator may enter. This value must be within the minimum and maximum entry values. If a value lower than this limit is entered the SOI will display an "Input Error" screen displaying the minimum and maximum entry limits.  |
| High User Input Limit  | The maximum entry value that an operator may enter. This value must be within the minimum and maximum entry values. If a value higher than this limit is entered the SOI will display an "Input Error" screen displaying the minimum and maximum entry limits.   |
| Default Value          | This parameter defines a default value that is displayed at the entry location of the display. An entry of Z defines no default value.   |

**Entry Screen  
Parameter for:  
ASCII Data**

|                 | Description  |
|-----------------|--|
| Register Number | The programmable controller data location operand and address.   |
| Character Count | The number of characters (2 characters for each 16 bit data location) to be displayed, up to a maximum of 20 characters. The initial byte of the location identified by the register number is displayed first, then the second byte, the first byte of the next higher sequential location, and so on. To display 20 characters, a sequential block of ten 16 bit locations is read by the SOI. |



**Entry Screen**  
**Parameter for:**  
**32-Bit Unsigned Integer &**  
**32-Bit BCD Data**

|                         | Description  |
|-------------------------|--|
| Register Number         | The programmable controller data location operand and address.   |
| Digits Right of Decimal | The number of digits to be placed to the right of the decimal.   |
| Digits Left of Decimal  | The number of digits to be placed to the left of the decimal.  |
| Low User Input Limit    | The minimum entry value that an operator may enter. This value must be within the range of 0 to +99,999,999. If a value lower than this is entered, the SOI will display an "Input Error" screen displaying the minimum and maximum entry limits.  |
| High User Input Limit   | The maximum entry value that an operator may enter. This value must be within the range of 0 to +99,999,999. If a value higher than this is entered, the SOI will display an "Input Error" screen displaying the minimum and maximum entry limits. |
| Default Value           | This parameter defines a default value that is displayed at the entry location of the display. An entry of Z defines no default.   |

*Note: For the 32-Bit BCD selections, scaling of data is not supported. The defined register number and the next higher sequential number locations will define the data value.*

## DeviceNet-Specific Configuration Parameters

This section describes the DeviceNet-specific parameters that the SOI-SPS programming software supports. These parameters may be modified by selecting the SOI Configuration Data menu option in the SOI-SPS software.

*Note: Refer to the SOI-SPS programming software manual for complete details on programming and additional information regarding the parameters accessible from the SOI Configuration Data menu.*

### Update Interval

This parameter defines the interval at which the SOI-120 or SOI-260 updates its screen data. The default value is 0.5 seconds. The range is 0.5 to 12.5 seconds.



### Time Synchronization (SOI-260 only)

The SOI-260's optional Real-Time Clock may either be written to the programmable controller or synchronized with the programmable controller. After selecting the Time Synchronization menu option from the SOI Configuration Data menu, select either the Write Time or Read Time menu option to enable this feature. You are then prompted for a base register.

The Real-Time Clock data is written to/read from the programmable controller, starting with the base register you enter. The Real-Time Clock data consists of six words:

- Word 1 Year, tens units (e.g., 95 = 1995)
- Word 2 Month, 1–12
- Word 3 Day of Month, 1-31
- Word 4 Hour (in military time), 0–23
- Word 5 Minutes, 0–59
- Word 6 Seconds, 0–59

Once this feature is enabled, the clock data is updated every 60 seconds.

### PLC Hardware Parameters

Select the PLC Hardware Parameters menu option from the SOI Configuration Data menu to modify the Communications Port Setup or the Advisor Update Rate.

The default **Communications Port Setup** is:

- Baud Rate: 125,000
- Node Address: 63
- Input Buffer Size: 1
- Output Buffer Size: 1
- Bus Off Interrupt: Hold in reset

The default **Advisor Update Rate** is 2 seconds.

*Note: If you change the communication port parameters, you must also change the corresponding parameters at the DeviceNet master (the scanner).*

*Note: You may also modify the communications port setup from the SOI-260. See “Chapter 4: Special Operations.”*



## CHAPTER 3: TROUBLESHOOTING

This chapter of the *DeviceNet Communications Reference* provides solutions to possible problems with the SOI-120 and SOI-260 and their use with DeviceNet. This chapter also contains a description of how to check the status of the system.

### General Symptoms and Solutions

Typically, any problems that occur involve communications between the SOI-120 or SOI-260 and the DeviceNet master (the scanner). The communications software used to communicate with the DeviceNet master has passed critical compliance tests, usually eliminating the software as a possible source of a problem.

The only other possible sources are the cabling and the communication port parameters involved (baud rate, node address/MAC ID, input buffer size, and output buffer size). The communication parameters must be identical for both the SOI-120 or SOI-260 and the DeviceNet master (the scanner). The cabling and the communications port parameters are always the first things to check.

*Note:* You can check the communications port parameters either from the SOI (see “Chapter 4: Special Operations”) or from the SOI-SPS programming software (see “Chapter 2: Programming”).

If the cabling and communications parameters are correct, put the SOI-120 or SOI-260 in Self-Test mode to rule out any non-functioning features of the SOI.

*Note:* Refer to the SOI-260 User’s Manual for detailed information about the Self-Test Mode.



The following screens appear on the SOI to indicate a problem:

**Network Power Lost**

24 Vdc is not present on the DeviceNet bus.

**No CANchip Reset  
"Y" to Reset**

The SOI-120/260 failed its start-up tests. Press "1" to reset the SOI-120 or press "Y" to reset the SOI-260, or cycle power to the SOI-120/260.

**CANchip Flt addr  
wr rd "Y"=Reset**

The SOI-120/260 failed to communicate with the CAN chip: addr = the hex address written to; wr = the data written; rd = the data read back. Press "1" to reset the SOI-120 or press "Y" to reset the SOI-260, or cycle power to the SOI-120/260.

**COMM Flt:BUS OFF  
"Y" key to Reset**

A bus-off interrupt occurred. Check the DeviceNet Communications Port Setup (see Chapter 4) to verify that the Baud Rate is consistent with all other devices on the bus. Press "1" to reset the SOI-120 or press "Y" to reset the SOI-260, or cycle power to the SOI-120/260.

**DUP MAC ID FAULT  
"Y" key for Info**

Some other device already owns your MAC ID (node address). Press "1" on the SOI-120 or "Y" on the SOI-260 to identify the device that owns the node address (see the screen below). You can use the DeviceNet Communications Port Setup menu (see Chapter 4) to change the node address from the SOI-120/260.

**MI VVVV SNSNSNSN**

After detecting a duplicate MAC ID (and if the user presses "1" or "Y" as described above), the SOI-120/260 displays this screen:

MI = the MAC ID

VVVV = the vendor ID

SNSNSNSN = the Electronic Serial Number (ESN)

The following screens appear on the SOI-120/260 during start-up and only indicate a problem if they last an inordinately long time:

**Confirming  
Node Address.**

If this screen lasts more than 2 seconds, the SOI-120/260's baud rate may be incorrect. Check the DeviceNet Communications Port Setup (see Chapter 4) to verify that the Baud Rate is consistent with all other devices on the bus.

**Waiting for  
Master Connection.**

If this screen lasts more than a few seconds, the DeviceNet master (the scanner) may not have the SOI-120/260 in its scan table. Refer to the scanner documentation.

**Waiting for  
I/O Connections**

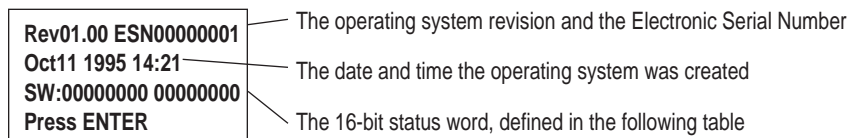
If this screen lasts more than 4 seconds, the SOI-120/260's input and output buffer sizes may be incorrect. Check the DeviceNet Communications Port Setup (see Chapter 4) to verify that the input and output buffer sizes are consistent with the sizes used to set up the DeviceNet master (the scanner).

If any of the above screens persist, you may have a hardware failure that requires service.



## System Status (SOI-260 only)

To check system status, press the “F” key on the SOI-260, select menu option “8 Other,” and then select menu option “7 Status.” A screen similar to the following should appear:



| Bit ① | Meaning  |
|-------|--|
| 0–2   | 000 = power loss occurred<br>001 = watchdog/power transient occurred<br>010 = user request occurred via Function (“F” key) menu<br>011 = software/hardware fault occurred<br>100 = reset request received from DeviceNet bus |
| 3     | 1 = Watchdog hardware failed   |
| 4     | 1 = Checksum error (operating system or configuration)   |
| 5     | 1 = RAM failure occurred   |
| 6     | 1 = Protocol built-in test failed  |
| 7     | 1 = Watchdog reset occurred  |
| 8     | 1 = Stack overflow   |
| 9     | Reserved   |
| 10    | 1 = Protocol message lost  |
| 11    | Reserved   |
| 12    | 1 = Protocol bus warning (there is a high error rate)  |
| 13    | 1 = Protocol bus off   |
| 14    | 1 = Protocol bus power loss  |
| 15    | 1 = Other DeviceNet bus failure (such as stuff error, form error, bit 0–1 errors, ack error, or CRC error)   |

① Bit 0 is the rightmost bit; Bit 15 is the leftmost bit.

Bits 0–7 are set at power-up. Bits 8–15 are set at runtime.



Pressing “CE” clears the runtime-detected faults (bits 8–15). This can help you determine if the faults are reoccurring. In general, if any of the faults persist, you may have a hardware failure that requires service.

*Note: You can also check the status of the DeviceNet from the SOI-260. See “Chapter 4: Special Operations.”*

## CHAPTER 4: SPECIAL OPERATIONS

After pressing “MODE” (the mode key) on the SOI-120 or “F” (the function key) on the SOI-260, one of the menu items is labelled “Special”. This item provides access to three features:

- *The P-A/D feature* gives you access to all unrestricted programmable controller addresses. With this access, addresses may be displayed and/or modified. This feature is extremely useful in start-up or debugging sessions of programs or regular programmable controller operations.
- *The DeviceNet communications port setup feature* gives you the ability to set up the DeviceNet communications port from the SOI-120 and SOI-260.
- *The DeviceNet status feature* gives you the ability to check the status of the DeviceNet from the SOI-120 and SOI-260.



**CAUTION:** The Point-Access/Display feature is a powerful tool. Its access should be restricted solely to authorized personnel. When using this feature, it is possible to change programmable controller data that may alter critical process control operations.

---

*Note: After pressing “MODE” (SOI-120) or “F” (SOI-260), a Special Security Screen may be displayed (it has to be created with the SOI-SPS Programming Software). You must enter one of the three programmed security codes to continue to the Special Operations menu screen.*

## Special Operations on the SOI-120

### Point-Access/Display

Figure 4.1 shows the Point-Access/Display screens and the key presses linking them:

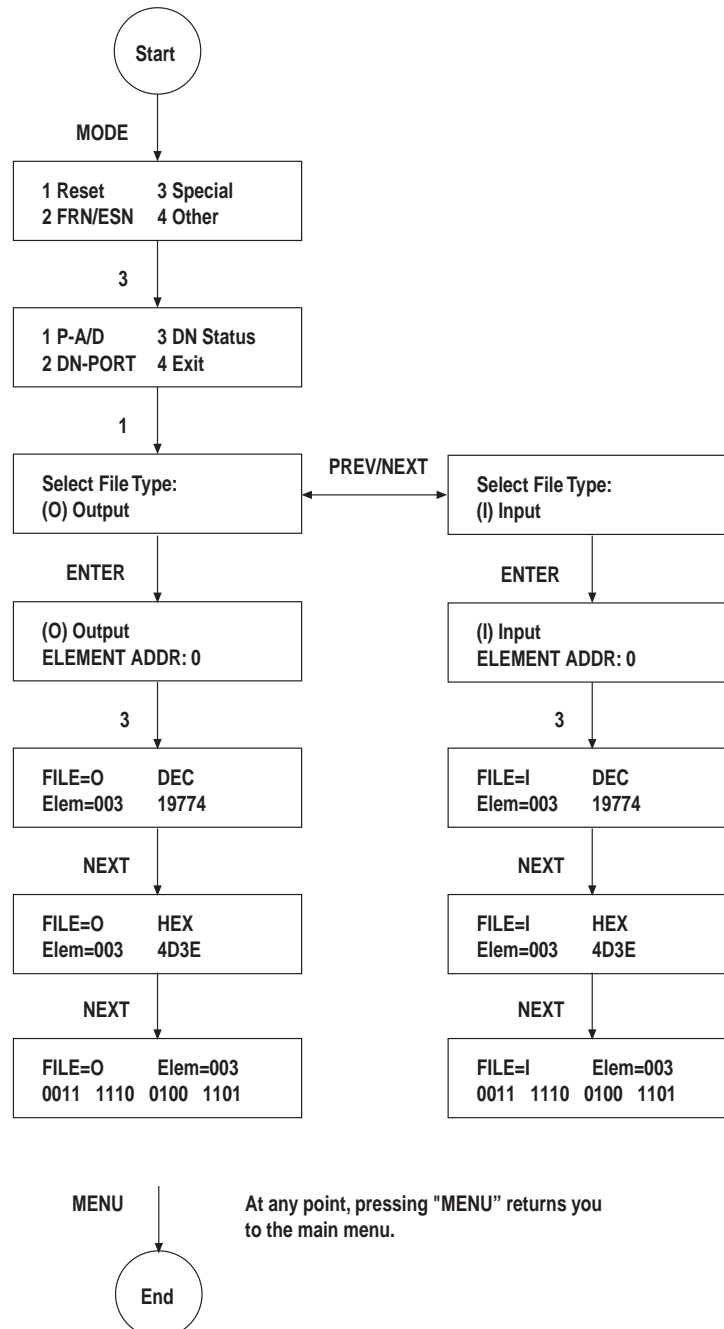


Figure 4.1 SOI-120 Point-Access/Display (P-A/D) screens



---

---

**To display or modify programmable controller addresses from the SOI-120:**

1. Press “MODE” (the mode key) on the SOI-120 keypad.
2. Select “Special” (item 3) from the menu that appears. If a Special Security Screen appears, enter any one of the programmed security codes.
3. Select “P-A/D” (item 1).
4. Press “PREV” or “NEXT” until the file type you want to display or modify appears, and then press “↵” (the enter key).
5. Enter the address you want to display or modify.
6. Press “PREV” or “NEXT” to move among decimal, hexadecimal, or binary display.
7. Press “↵” (the enter key) to load new data (Input file types only) into the programmable controller. This new data is displayed immediately.

*Note: At any point, pressing “MENU” returns you to the main menu.*

---

---

### DeviceNet Communications Port Setup

Figure 4.2 shows the DeviceNet Communications Port Setup screens and the key presses linking them:

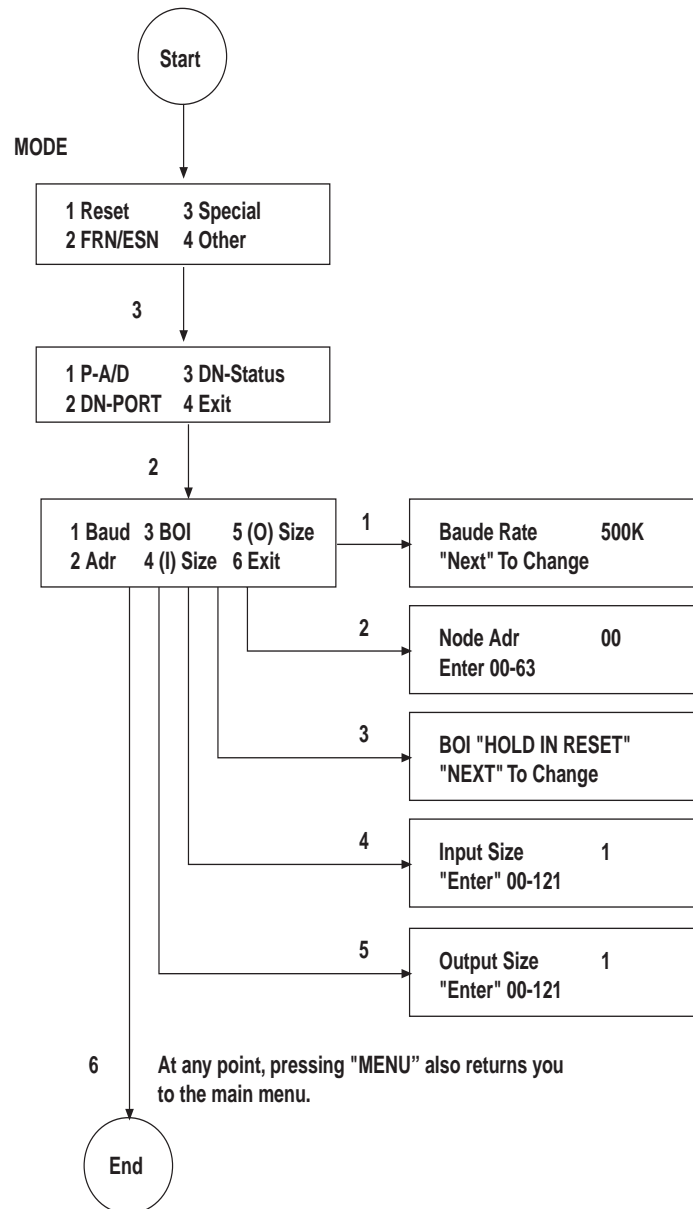


Figure 4.2 SOI-120 DeviceNet communications port setup screens.

---

#### To set up the DeviceNet port from the SOI-120:

1. Press "MODE" (the mode key) on the SOI-120 keypad.



2. Select “Special” (item 3) from the menu that appears. If a Special Security Screen appears, enter any one of the programmed security codes.
3. Select “DN-PORT” (item 2).
4. Select the DeviceNet Communications Port parameter you want to display or modify.
5. For the “Baud” (baud rate) and “BOI” (bus off interrupt) parameters, press “PREV” or “NEXT” to cycle through the available settings. For the “Adr” (MAC ID) parameter, enter the node address (0–63). For the “(I) Size” (input buffer size) and “(O) Size” (output buffer size) parameters, enter the desired buffer size (0–121).

*Note: If you change the communication port parameters, you must also change the corresponding parameters at the DeviceNet master (the scanner).*

6. Press “↵” (the enter key) to continue.

*Note: At any point, pressing “MENU” returns you to the main menu.*

---

### DeviceNet Status

Figure 4.3 shows the DeviceNet Status screens and the key presses linking them:

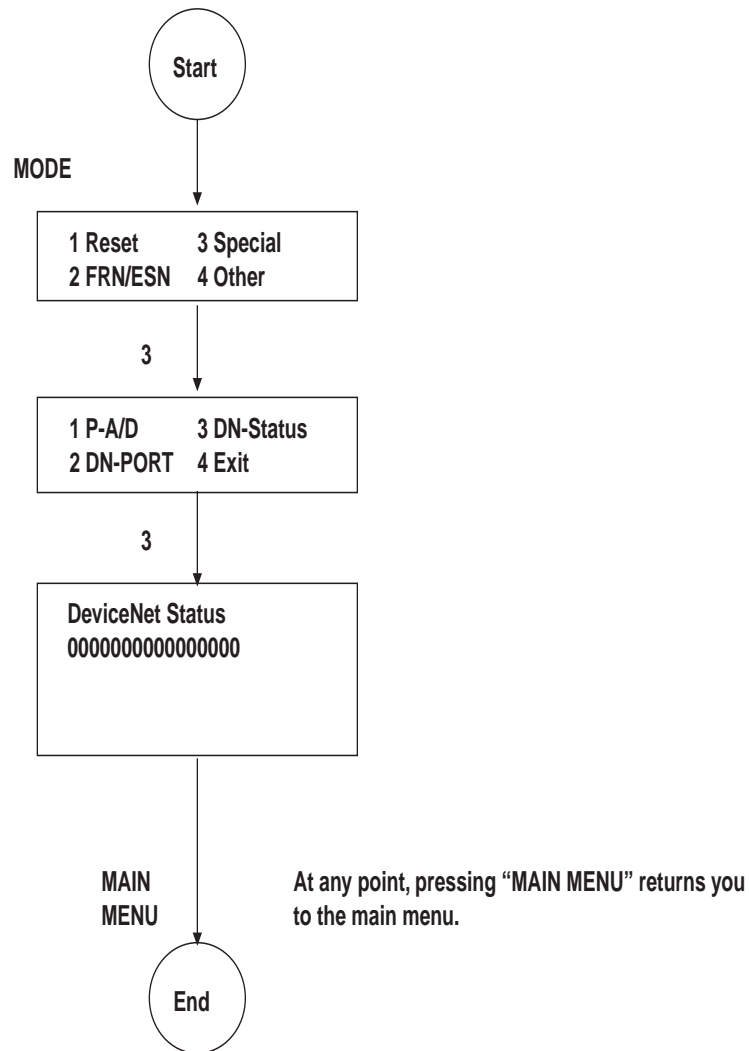


Figure 4.3 SOI-120 DeviceNet Status screens.

---

#### To check the DeviceNet status from the SOI-120:

1. Press "MODE" (the mode key) on the SOI-120 keypad.
2. Select "Special" (item 3) from the menu that appears. If a Special Security Screen appears, enter any one of the programmed security codes.



3. Press “3” to view the DeviceNet Status screen. Line 2 shows the DeviceNet Status word, defined in the following table:

| Bit ① | Meaning  |
|-------|--|
| 0     | 1 = Master/slave connection set has been allocated                   |
| 1     | Reserved   |
| 2     | 1 = Configuration parameters have been changed from factory defaults |
| 3     | Reserved   |
| 4–7   | Not used   |
| 8     | 1 = Minor recoverable device fault                                   |
| 9     | 1 = Minor unrecoverable device fault                                 |
| 10    | 1 = Major recoverable device fault                                   |
| 11    | 1 = Major unrecoverable device fault                                 |
| 12–15 | Reserved   |

① Bit 0 is the rightmost bit; Bit 15 is the leftmost bit.

*Note: At any point, pressing “MENU” returns you to the main menu.*

## Special Operations on the SOI-260

### Point-Access/Display

Figure 4.4 shows the Point-Access/Display screens and the key presses linking them:

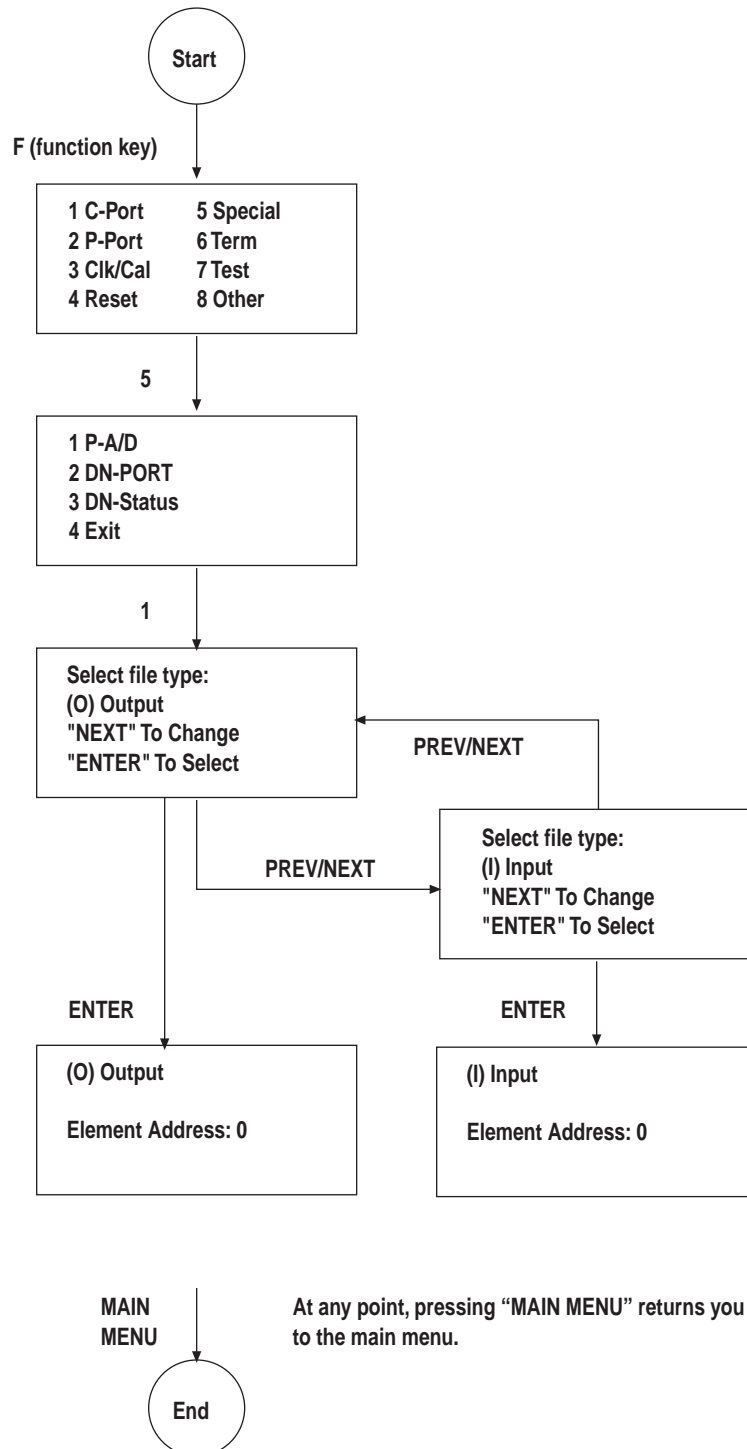


Figure 4.4 SOI-260 Point-Access/Display (P-A/D) screens

---

**To display or modify programmable controller addresses from the SOI-260:**

1. Press “F” (the function key) on the SOI-260 keypad.



2. Select “Special” (item 5) from the menu that appears. If a Special Security Screen appears, enter any one of the programmed security codes.
3. Press “1” to use the P-A/D function.
4. Press “PREV” or “NEXT” until the file type you want to display or modify appears, and then press “↵” (the enter key).
5. Enter the address you want to display or modify.
6. Press “PREV” or “NEXT” to move the cursor to the field you want to modify, and enter new data, if desired. Press “Y” or “N” to increment or decrement the address shown.
7. Press “↵” (the enter key) to load the new data into the programmable controller. The new data that was loaded into the programmable controller is immediately displayed.

*Note: At any point, pressing “MAIN MENU” returns you to the main menu.*

---

---

### DeviceNet Communications Port Setup

Figure 4.5 shows the DeviceNet Communications Port Setup screens and the key presses linking them:

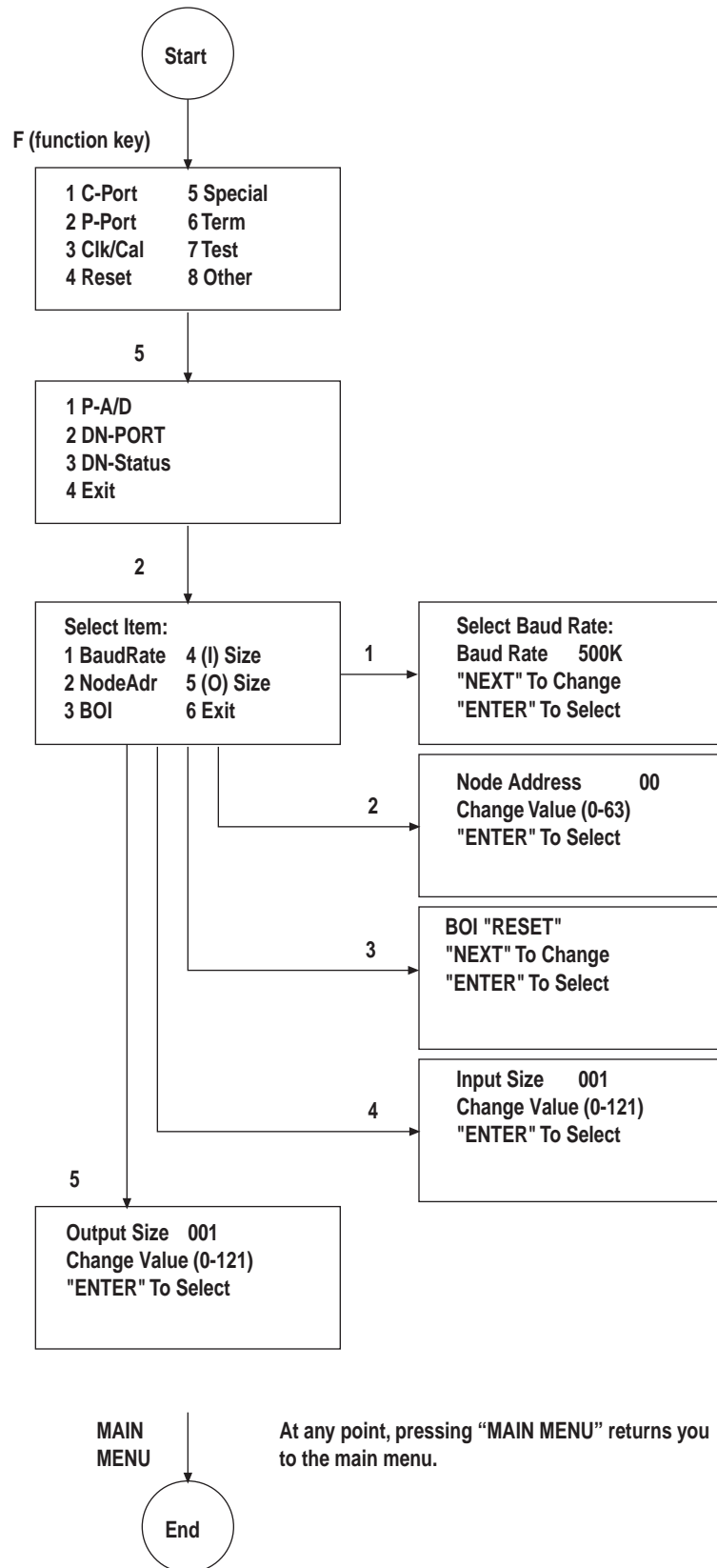


Figure 4.5 SOI-260 DeviceNet communications port setup screens.



---

---

**To set up the DeviceNet port from the SOI-260:**

1. Press “F” (the function key) on the SOI-260 keypad.
2. Select “Special” (item 5) from the menu that appears. If a Special Security Screen appears, enter any one of the programmed security codes.
3. Press “2” to use the DeviceNet Communications Port Setup feature.
4. Select the DeviceNet Communications Port parameter you want to display or modify.
5. For the “BaudRate” and “BOI” (bus off interrupt) parameters, press “PREV” or “NEXT” to cycle through the available settings. For the “NodeAdr” (MAC ID) parameter, enter the node address (0–63). For the “(I) Size” (input buffer size) and “(O) Size” (output buffer size) parameters, enter the desired buffer size (1–121).

*Note: If you change the communication port parameters, you must also change the corresponding parameters at the DeviceNet master (the scanner).*

6. Press “↵” (the enter key) to continue.

*Note: At any point, pressing “MAIN MENU” returns you to the main menu.*

---

---

### DeviceNet Status

Figure 4.6 shows the DeviceNet Status screens and the key presses linking them:

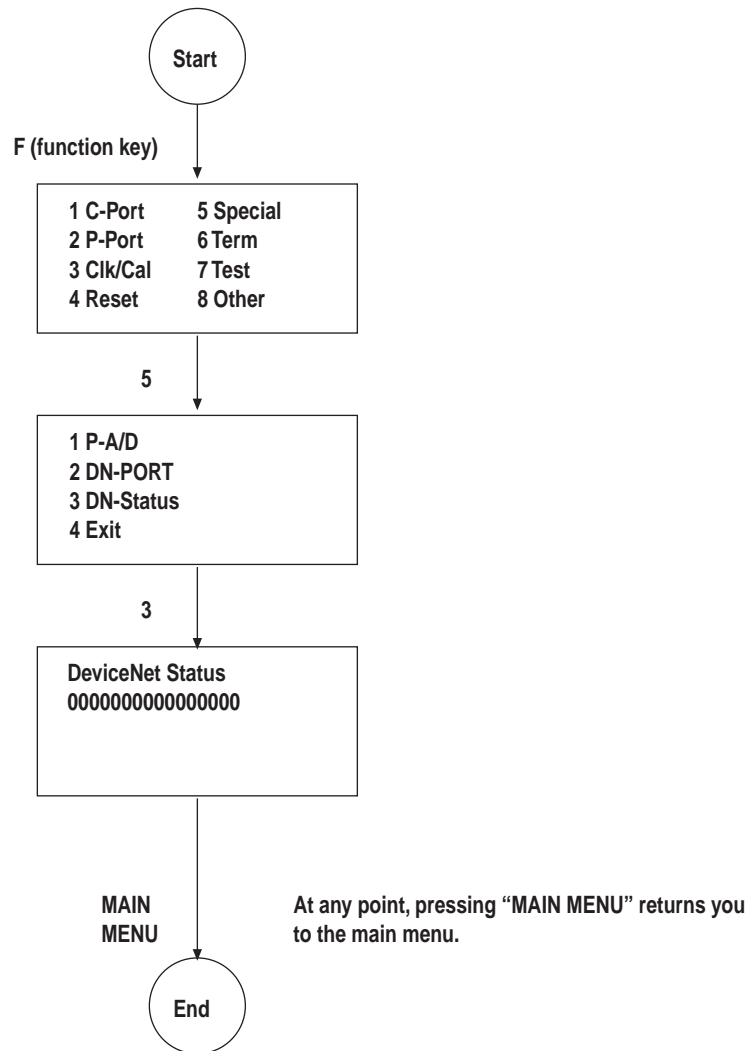


Figure 4.6 SOI-260 DeviceNet Status screens.

---

#### To check the DeviceNet status from the SOI-260:

1. Press "F" (the function key) on the SOI-260 keypad.
2. Select "Special" (item 5) from the menu that appears. If a Special Security Screen appears, enter any one of the programmed security codes.



3. Press “3” to view the DeviceNet Status screen. Line 2 shows the DeviceNet Status word, defined in the following table:

| Bit ① | Meaning  |
|-------|--|
| 0     | 1 = Master/slave connection set has been allocated                   |
| 1     | Reserved   |
| 2     | 1 = Configuration parameters have been changed from factory defaults |
| 3     | Reserved   |
| 4–7   | Not used   |
| 8     | 1 = Minor recoverable device fault                                   |
| 9     | 1 = Minor unrecoverable device fault                                 |
| 10    | 1 = Major recoverable device fault                                   |
| 11    | 1 = Major unrecoverable device fault                                 |
| 12–15 | Reserved   |

① Bit 0 is the rightmost bit; Bit 15 is the leftmost bit.

**Note:** *At any point, pressing “MAIN MENU” returns you to the main menu.*



## APPENDIX A: DEVICENET SPECIFICATIONS

A DeviceNet communications cable is required to allow the SOI-120 and SOI-260 to connect to the DeviceNet (see “Chapter 1: Cabling and Communication Ports”).

### Electrical Specifications

|                      |   |
|----------------------|---|
| DeviceNet baud rate  | 125, 250, or 500 kbps   |
| Termination resistor | 121 $\Omega$ , 1% metal film, 1/4 watt, required at each end of the trunk |

### Physical Specifications

|                              |  |
|------------------------------|--|
| SOI-120 depth with DeviceNet | 1.8 in. max.   |
| SOI-260 depth with DeviceNet | 2.5 in. max.   |
| Trunk distance               | 500 meters at 125 kbps<br>200 meters at 250 kbps<br>100 meters at 500 kbps |



## APPENDIX B: DEVICE PROFILE

The SOI-120 and SOI-260 work as slaves “owned” by a single master (the scanner), as described in Figure B.1.

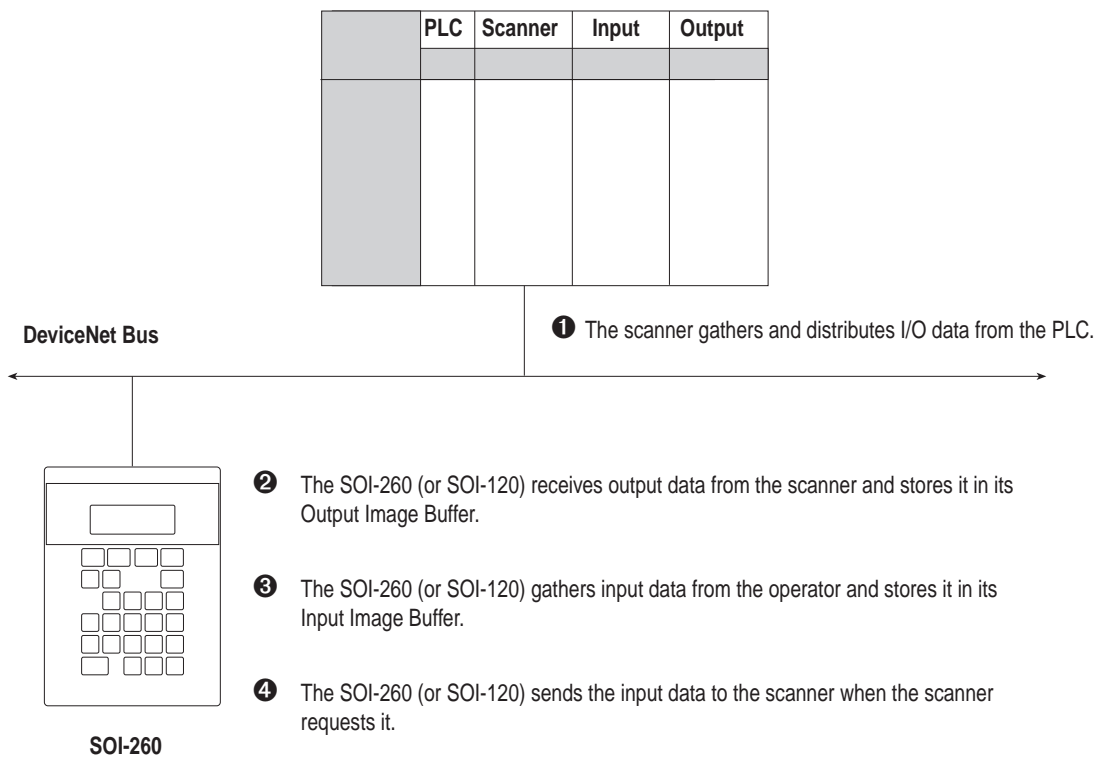


Figure B.1 Typical DeviceNet Configuration.



As defined by the Open DeviceNet Vendors Association, the SOI-120 and SOI-260 are Unconnected Message Manager (UCMM) capable and operate as a Group 2 server (not a Group 2 *only* server). This eliminates the burden of having a separate device operating as the Group 2 only client and performing the UCMM services for the SOI-260. As a slave, the SOI-120 and SOI-260 support I/O polled messages, but not I/O bit strobe messages. To allow transmission of more than 8 data bytes, the SOI-120 and SOI-260 support the fragmentation protocol.

The rest of this appendix defines the DeviceNet message types, class services, and objects that are supported by the SOI-120 and SOI-260.

### DeviceNet Message Types

As an Unconnected Message Manager (UCMM) capable Group 2 Server acting as a slave, the SOI-120 and SOI-260 support the following group 1, group 2, and group 3 messages:

| 10                | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Message ID/Meaning                     |
|-------------------|---|---|---|---|---|---|---|---|---|---|--|
| Group 1 Message:  |   |   |   |   |   |   |   |   |   |   |  |
| 0                 | 1 | 1 | 1 | 1 | X | X | X | X | X | X | Slave's I/O Poll Response Message      |
| Group 2 Messages: |   |   |   |   |   |   |   |   |   |   |  |
| 1                 | 0 | X | X | X | X | X | X | 0 | 1 | 1 | ② Slave's Explicit Response Messages   |
| 1                 | 0 | X | X | X | X | X | X | 1 | 0 | 0 | ② Master's Explicit Request Messages   |
| 1                 | 0 | X | X | X | X | X | X | 1 | 0 | 1 | Master's I/O Poll Command Message      |
| 1                 | 0 | X | X | X | X | X | X | 1 | 1 | 1 | Duplicate MAC ID Check Messages        |
| Group 3 Messages: |   |   |   |   |   |   |   |   |   |   |  |
| 1                 | 1 | 1 | 0 | 1 | X | X | X | X | X | X | Unconnected Explicit Response Message  |
| 1                 | 1 | 1 | 1 | 0 | Y | Y | Y | Y | Y | Y | ① Unconnected Explicit Request Message |

X X X X X X = SOI-120 or SOI-260 Node Address  
 Y Y Y Y Y Y = DeviceNet Master Node Address

- ① The SOI-120 and SOI-260 can receive but not send Unconnected Explicit Request Messages.
- ② The SOI-120 and SOI-260 support only the 8/8 Explicit Message Connection Data Size. (8/16, 16/16, and 16/8 will not be supported.)

Other unused message IDs not listed above are available in the message pools for UCMM connections.



## DeviceNet Object Classes

The SOI-120 and SOI-260 support the following DeviceNet object classes:

| Class Code: | Number of Instances: | Comments:                    |
|-------------|----------------------|------------------------------|
| 01 hex      | 1                    | Identity Object              |
| 02 hex      | 1                    | Message Router Object        |
| 03 hex      | 1                    | DeviceNet Object             |
| 04 hex      | 2                    | Assembly Objects             |
| 05 hex      | 4                    | DeviceNet Connection Objects |
| 64 hex      | 1                    | OI Application Object        |

## Identity Object

**Class Code: 01hex**

**Number of Instances = 1**

**Class Attributes:**

| Attribute ID | Access Rule | Name     | DeviceNet Data Type | Default Data Value |
|--------------|-------------|----------|---------------------|--------------------|
| 1            | Get         | Revision | UINT                | 1                  |



**Instance Attributes:**

| Attribute ID | Access Rule | Name   | DeviceNet Data Type            | Default Data Value         |
|--------------|-------------|--|--------------------------------|----------------------------|
| 1            | Get         | Vendor ID                                      | UINT                           | 58                         |
| 2            | Get         | Device Type                                    | UINT                           | 0=generic device           |
| 3            | Get         | Product Code                                   | UINT                           | 1 = SOI-260<br>2 = SOI-120 |
| 4            | Get         | Revision:<br>Major Revision<br>Minor Revision  | struct of<br>USINT<br>USINT    | 1<br>0                     |
| 5            | Get         | Status   | WORD                           | device status*             |
| 6            | Get         | Electronic<br>Serial Number ①                  | UDINT                          | unique 32bit number        |
| 7            | Get         | Product Name:<br>String Length<br>ASCII String | struct of<br>USINT<br>STRING[] | 6,"SOI260"<br>7,"SOI120"   |

① This is a programmable configuration item.

**Common Services:**

| Service Code<br>(Hex) | Implemented for: |          | Service Name         |
|-----------------------|------------------|----------|----------------------|
|                       | Class            | Instance |                      |
| 0E                    | Yes              | Yes      | Get_Attribute_Single |
| 05                    | No               | Yes      | Reset                |



\*device status:

| Bit(s) | Name                      | Definition  |
|--------|---------------------------|---|
| 0      | owned                     | 0=not owned<br>1=owned (Master/Slave Connection set has been allocated) |
| 1      | reserved                  | set to 0  |
| 2      | Configured                | 0=Out of Box CFG<br>1=Not Out of box Cfg                                |
| 3      | reserved                  | set to 0  |
| 4-7    | vendor specific           | 0 = not used  |
| 8      | Minor Recoverable Fault   | 0 = no fault<br>1 = minor recoverable device fault                      |
| 9      | Minor Unrecoverable Fault | 0 = no fault<br>1 = minor unrecoverable device fault                    |
| 10     | Major Recoverable Fault   | 0 = no fault<br>1 = major recoverable device fault                      |
| 11     | Major Unrecoverable Fault | 0 = no fault<br>1 = major unrecoverable device fault                    |
| 12-13  | reserved                  | set to 0  |
| 14-15  | reserved                  | set to 0  |

## Out-Of-Box Definition

The parameters and the defaults listed in the following table define the “Out-of-Box” configuration for the SOI-120 and SOI-260 DeviceNet products. These are set at the factory and may also be set through the DeviceNet Identity object explicit message reset function.



| <b>Function:</b>        | <b>Parameter:</b> | <b>Default Value:</b>   |
|-------------------------|-------------------|-------------------------|
| serial port             | baud rate         | (Not used)              |
| serial port             | line cfg          | (Not used)              |
| printer port*           | baud rate         | 9600 Baud               |
| printer port*           | line cfg          | 8 data bits Even parity |
| simulate                | enable            | 0=OFF                   |
| backlight               | enable            | 1=ON                    |
| contrast*               | enable            | 1=ON                    |
| printer port*           | enable            | 1=Enabled               |
| printer port LF*        | enable            | 0=Disabled              |
| printer port handshake* | enable            | 0=Disabled              |
| print screen "E" key*   | enable            | 1=Disabled              |
| scale                   | enable            | 1=Enabled               |
| DeviceNet               | boi action        | 0 = hold in reset.      |
| DeviceNet               | input file size   | 1 word                  |
| DeviceNet               | output file size  | 1 word                  |
| DeviceNet               | node address      | 63 dec                  |
| DeviceNet               | Baud Rate         | 125 KBaud               |

\* SOI-260 only.



## Major/Minor Fault Assignments

| <b>Minor Recoverable Faults:</b>   | <b>Detection (power up or runtime):</b> |
|------------------------------------|---|
| DeviceNet Bus Warning              | Runtime                                 |
| DeviceNet CRC Error                | Runtime                                 |
| DeviceNet Bit 0 Error              | Runtime                                 |
| DeviceNet Bit 1 Error              | Runtime                                 |
| DeviceNet Stuff Error              | Runtime                                 |
| DeviceNet Format Error             | Runtime                                 |
| DeviceNet Ack Error                | Runtime                                 |
| <b>Minor Unrecoverable Faults:</b> | <b>Detection (power up or runtime):</b> |
| none                               | NA                                      |
| <b>Major Recoverable Faults:</b>   | <b>Detection (power up or runtime):</b> |
| DeviceNet Bus Off                  | Power up                                |
| Watchdog/Pwr transient             | Power up                                |
| Stack overflow                     | Power up                                |
| Checksum Fail (OS or CFG)          | Power up                                |
| RAM BITE Fail                      | Power up                                |
| Protocol HW BITE fail              | Power up                                |
| <b>Major Unrecoverable Faults:</b> | <b>Detection By:</b>                    |
| none                               | NA                                      |



## Message Router Object

Class Code: 02hex

Number of Instances = 1

### Class Attributes:

| Attribute ID | Access Rule | Name     | DeviceNet Data Type | Default Data Value |
|--------------|-------------|----------|---------------------|--------------------|
| 1            | Get         | Revision | UINT                | 1                  |

### Instance Attributes:

| Attribute ID | Access Rule | Name  | DeviceNet Data Type                    | Default Data Value     |
|--------------|-------------|---|--|------------------------|
| 1            | Get         | Object_list<br>Number of class<br>codes supports<br>Classes | STRUCT of<br><br>UINT<br>ARRAY of UINT | 5<br><br>1,2,3,4,5 hex |

### Common Services:

| Service Code<br>(Hex) | Implemented for: |          | Service Name         |
|-----------------------|------------------|----------|----------------------|
|                       | Class            | Instance |                      |
| 0E                    | Yes              | Yes      | Get_Attribute_Single |



## DeviceNet Object

Class Code: 03hex

Number of Instances = 1

### Class Attributes:

| Attribute ID | Access Rule | Name     | DeviceNet Data Type | Default Data Value |
|--------------|-------------|----------|---------------------|--------------------|
| 1            | Get         | Revision | UINT                | 1                  |

### Instance Attributes:

| Attribute ID | Access Rule | Name            | DeviceNet Data Type      | Default Data Value                                     |
|--------------|-------------|-----------------|--------------------------|--|
| 1            | Get         | ① Node Address  | USINT                    | 0-63   |
| 2            | Get         | ① Baud Rate     | USINT                    | 0=125K<br>1=250K<br>2=500K                             |
| 3            | Get/Set     | ① BOI           | BOOL                     | default=0  |
| 4            | Get/Set     | Bus-Off Counter | USINT                    | 0-255*   |
| 5            | Get         | Allocation Info | Struct:<br>BYTE<br>USINT | **allocation byte<br>0-63=master ID<br>255=unallocated |

① This is a programmable configuration item.



**Common Services:**

| Service Code<br>(Hex) | Implemented for: |          | Service Name                         |
|-----------------------|------------------|----------|--------------------------------------|
|                       | Class            | Instance |                                      |
| 0E                    | Yes              | Yes      | Get_Attribute_Single                 |
| 10                    | No               | Yes      | Set_Attribute_Single                 |
| 4B                    | No               | Yes      | Allocate Master/Slave Connection Set |
| 4C                    | No               | Yes      | Release Master/Slave Connection Set  |

\*Set\_Attribute\_Single service for instance attribute 4 will always set BOI to 0 regardless of data value. The BOI counter can not be preset to any value other than 0.

\*\*Allocation byte:

| Bit(s) | Name       | Default Data Value                                  |
|--------|------------|---|
| 7-3    | reserved   | Always 0  |
| 2      | bit_strobe | Not supported. Always 0.                            |
| 1      | Polled     | 0-not requested<br>1-requesting polled allocation   |
| 0      | Explicit   | 0-not requested<br>1-requesting explicit allocation |



## Connection Objects

The DeviceNet connections supported for the Master/Slave implementation are as follows:

| Connection Instance ID# | Description                                 |
|-------------------------|---|
| 1                       | Predefined Master/Slave Explicit Connection |
| 2                       | Predefined Master/Slave Poll I/O Connection |
| 3                       | Primary UCMM Explicit Connection            |
| 4                       | Secondary UCMM Explicit Connection          |

The Explicit Connections are responsible for receiving any explicit requests and returning the associated responses.

The Poll Connection is responsible for receiving the Master's Poll Command and returning the associated Poll Response.

**Class Code: 05hex**

**Number of Instances = 4**

**Class Attributes:**

| Attribute ID | Access Rule | Name     | DeviceNet Data Type | Default Data Value |
|--------------|-------------|----------|---------------------|--------------------|
| 1            | Get         | Revision | UINT                | 1                  |



**Instance 1,3, and 4 Attributes (Explicit Message Connections):**

| Attribute ID | Access Rule | Name   | DeviceNet Data Type | Default Data Value                        |
|--------------|-------------|--|---------------------|---|
| 1            | Get         | State  | USINT               | 03 - Established                          |
| 2            | Get         | Instance Type                                | USINT               | 00-Explicit                               |
| 3            | Get         | Transport Trigger                            | USINT               | 83hex:<br>Server/Transport class 3        |
| 4            | Get         | Produced<br>Connection ID<br>(Slave Exp Rsp) | UINT                | 10xxxxxx011binary<br>xxxxxx = nodeaddress |
| 5            | Get         | Consumed<br>Connection ID<br>(Master Exp Rq) | UINT                | 10xxxxxx100binary<br>xxxxxx = nodeaddress |
| 6            | Get         | Initial<br>Comm characteristics              | USINT               | 21hex                                     |
| 7            | Get         | Produced<br>Connection Size                  | UINT                | 7   |
| 8            | Get         | Consumed<br>Connection Size                  | UINT                | 7   |
| 9            | Get/Set     | Expect<br>Packet Rate                        | UINT                | 09C4hex                                   |
| 12           | Get         | Watchdog<br>Timeout Action                   | USINT               | 1=auto delete                             |
| 13           | Get         | Produced<br>Connection Path Length           | UINT                | 0   |
| 14           | Get         | Produced<br>Connection Path                  | Array of USINT      | empty                                     |
| 15           | Get         | Consumed<br>Connection Path Length           | UINT                | 0   |
| 16           | Get         | Consumed<br>Connection Path                  | Array of USINT      | empty                                     |



### Instance 2 Attributes (Poll I/O Connection):

| Attribute ID | Access Rule | Name   | DeviceNet Data Type | Default Data Value                        |
|--------------|-------------|--|---------------------|---|
| 1            | Get         | State  | USINT               | 01 - Configuring                          |
| 2            | Get         | Instance Type                                  | USINT               | 01-I/O                                    |
| 3            | Get         | Transport Trigger                              | USINT               | 82hex:<br>Server/Transport class 2        |
| 4            | Get         | Produced<br>Connection ID<br>(Slave I/O Poll)  | UINT                | 01111xxxxxbinary<br>xxxxxx = nodeaddress  |
| 5            | Get         | Consumed<br>Connection ID<br>(Master I/O Poll) | UINT                | 10xxxxxx101binary<br>xxxxxx = nodeaddress |
| 6            | Get         | Initial<br>Comm characteristics                | USINT               | 01hex                                     |
| 7            | Get         | 1Produced<br>Connection Size                   | UINT                | input_file_size                           |
| 8            | Get         | 1Consumed<br>Connection Size                   | UINT                | output_file_size                          |
| 9            | Get/Set     | Expect<br>Packet Rate                          | UINT                | 0   |
| 12           | Get         | Watchdog<br>Timeout Action                     | USINT               | 0=time out                                |
| 13           | Get         | Produced<br>Connection Path Length             | UINT                | 4   |
| 14           | Get         | Produced<br>Connection Path                    | Array of USINT      | Class 4<br>Instance 1<br>20 04 24 01      |
| 15           | Get         | Consumed<br>Connection Path Length             | UINT                | 4   |
| 16           | Get         | Consumed<br>Connection Path                    | Array of USINT      | Class 4<br>Instance 2<br>20 04 24 02      |



**Common Services:**

| Service Code<br>(Hex) | Implemented for: |          | Service Name         |
|-----------------------|------------------|----------|----------------------|
|                       | Class            | Instance |                      |
| 05                    | Yes              | Yes      | Reset*               |
| 0E                    | Yes              | Yes      | Get_Attribute_Single |
| 10                    | No               | Yes      | Set_Attribute_Single |

\*For class, resets all connections to non-existent. For instances, resets connection timer, and if applicable changes state from timed-out to established.

**Assembly Objects**

**Class Code: 04hex**

**Number of Instances = 2**

**Class Attributes:**

| Attribute ID | Access Rule | Name     | DeviceNet Data Type | Default Data Value |
|--------------|-------------|----------|---------------------|--------------------|
| 1            | Get         | Revision | UINT                | 1                  |

**Instance 1 Attributes (Input Assembly):**

| Attribute ID | Access Rule | Name | DeviceNet Data Type | Default Data Value |
|--------------|-------------|------|---------------------|--------------------|
| 3            | Get         | Data | Array of BYTE       | input_data         |



### Instance 2 Attributes (Output Assembly):

| Attribute ID | Access Rule | Name | DeviceNet Data Type | Default Data Value |
|--------------|-------------|------|---------------------|--------------------|
| 3            | Get         | Data | Array of BYTE       | output_data        |

### Common Services:

| Service Code<br>(Hex) | Implemented for: |          | Service Name          |
|-----------------------|------------------|----------|-----------------------|
|                       | Class            | Instance |                       |
| 0E                    | Yes              | Yes      | Get_Attribute_Single* |

\*Since the assembly objects may be up to 242 bytes each, acknowledged fragmented I/O must be supported for this explicit request function.

## Application Object

The application object is not accessible via the DeviceNet network.

**Class Code: 64hex**

**Number of Instances = 1**

### Class Attributes:

| Attribute ID         | Access Rule | Name | DeviceNet Data Type | Default Data Value |
|----------------------|-------------|------|---------------------|--------------------|
| no public attributes |             |      |                     |                    |



**Instance Attributes:**

| Attribute ID         | Access Rule | Name | DeviceNet Data Type | Default Data Value |
|----------------------|-------------|------|---------------------|--------------------|
| no public attributes |             |      |                     |                    |

**Common Services:**

| Service Code<br>(Hex) | Implemented for:<br>Class | Instance | Service Name |
|-----------------------|---------------------------|----------|--------------|
| no public services    |                           |          |              |



# INDEX

## SYMBOLS

- 16-bit formats
  - BCD 16
  - Hex 17
  - signed integer 15
  - unsigned integer 16
- 32-bit
  - floating point 19
- 32-bit formats
  - BCD 18
  - Hex 18
  - unsigned integer 17

## A

- Adapters 9
- Addresses, accessing 34–39, 39–45
- Advisor update rate 27
- Application programs, uploading/downloading 6
- ASCII
  - data input 11
  - display unit 11
  - format 19
  - input device 11

## B

- Barcode scanner 11
- Baud Rate 37, 43
- Baud rate 27, 47
- BCD formats
  - 16-bit 16
  - 32-bit 18
- Binary Coded Decimal. *See* BCD formats

- Bit format 15
- Buffer size
  - input 27, 37, 43
  - output 27, 37, 43
- Bus-off interrupt 27, 37, 43

## C

- Cabling 5
- Client 50
- Clock 27
- Communication
  - cabling 5
  - port setup 27, 36–39, 41–45
  - ports 5
  - software 29–30
- Computer. *See* Programming computer
- Controller. *See* Network master computer

## D

- Data
  - display 20–26
  - entry 20–26
  - formats supported 13–19, 20
- Default parameter settings 53–54
- Depth with DeviceNet 47
- DeviceNet
  - client 50
  - communication port setup 33, 36–39, 41–45
  - configuration 26
  - master 49
  - message types supported 50
  - object classes supported 51
  - parameters 26



- port 10
- server 50
- slave 49
- specifications 47
- status 33, 38, 44
- termination 10
- Display screen parameters 21–23
- Downloading applications
  - via optional printer port 7
  - via RS-232 communications port 6

## E

- Electronic serial number 31
- Electrostatic damage i
- Entry screen parameters 24–26
- Error messages 30, 31, 38, 44

## F

- Fault messages 30, 31, 38, 44
- File types supported 13
- Formats supported 13–19, 20
- Function key. *See* F key

## H

- Hardware fault 31, 38, 44
- Hex formats
  - 16-bit 17
  - 32-bit 18
- Host controller. *See* Network master computer

## I

- Input
  - buffer size 13, 27, 37, 43
  - file 13
  - image buffer 49
- Integer formats
  - 16-bit signed 15
  - 16-bit unsigned 16
  - 32-bit unsigned 17
- Interrupt 27

## L

- Liability i

## M

- Master 49
- Message types supported 50

## N

- Node address 27, 37, 43

## O

- Object classes supported 51
- Output
  - buffer size 13, 27, 37, 43
  - file 13
  - image buffer 49

## P

- P-A/D. *See* Point access/display
- Packaging instructions i
- Parameter settings, default 53–54
- Password 33
- Personal computer. *See* Programming computer
- Pin connections 5
- Point access/display 33, 34–39, 39–45
- Ports 5
  - communications setup 36–39, 41–45
- Printer forms, outputting 11
- Printer port 11
- Programming 13–27
- Programs, uploading/downloading 6

## R

- Real-time clock 27

## S

- Scanner 49
- SCC-3 cable 8
- Screen. *See* Display
- Screens
  - data
    - display 21–23
    - entry 24–26
    - security 33
    - types supported 20
- Security code 33
- Security screen 33
- Serial number, electronic 31–32
- Server 50
- Slave 49
- Special Operations 33–45
- Special security screen 33
- Specifications, DeviceNet 47
- Status
  - DeviceNet 38, 44
  - system 31
- Synchronization 27
- System status 31



## T

Terminal mode 2  
Termination resistor 47  
Time synchronization 27  
Troubleshooting 29–32  
Trunk distance 47

## U

Update interval 26  
Uploading applications  
    via optional printer port 7  
    via RS-232 communications port 6  
Uploading/downloading applications 6

## W

Warranty information ii





Copyright © 1998 Spectrum Controls, Inc. All rights reserved.  
Specifications subject to change without notice. Printed in U.S.A.  
Publication 0300142-03 Rev. C.0 April 1998

**Corporate Headquarters**

Spectrum Controls, Inc.  
P.O. Box 5533 • Bellevue, Washington 98006  
Fax: (425) 641-9473 • Tel: (425) 746-9481

**Northeastern U.S.A. Sales Office**

Spectrum Controls, Inc.  
48945 Van Dyke, 4B • Utica, Michigan 48317  
Fax: (586) 731-2715 • Tel: (586) 731-2397

**Southeastern U.S.A. Sales Office**

Spectrum Controls, Inc.  
8860 Saddle Trail • Ball Ground, Georgia 30107  
Fax: (678) 455-4615 • Tel: (678) 455-4640

Web Site: <http://www.spectrumcontrols.com>  
E-mail: [spectrum@spectrumcontrols.com](mailto:spectrum@spectrumcontrols.com)

